HDAT 9700

Time series analysis

Online tutorial

Overview

1. Understanding time series and working with time series data in R

2. Statistical properties of time series data

3. Introduction to time series models (ARIMA)

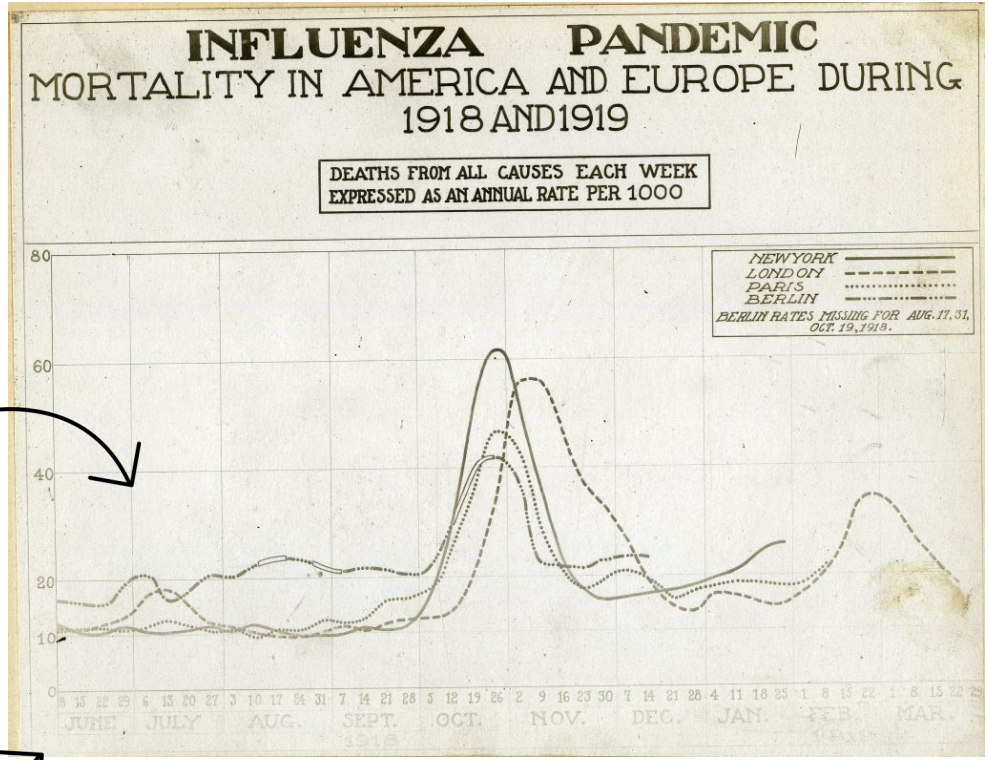
What are time series data?

- Chronological sequence of measurements equally spaced through time—**the order is important**
- One observation (per outcome) for every time point—often a **summary statistic**
- The **frequency is always the same** (e.g. daily, weekly, monthly, etc)
- **Important—observations made at neighbouring time points are often correlated and thus not independent

	month	dispensings
1	1-Jan-11	16831
2	1-Feb-11	17234
3	1-Mar-11	20546
4	1-Apr-11	19226
5	1-May-11	21136
6	1-Jun-11	20939
7	1-Jul-11	21103
8	1-Aug-11	22897
9	1-Sep-11	22162
10	1-Oct-11	22184
11	1-Nov-11	23108
12	1-Dec-11	25967
13	1-Jan-12	20123
14	1-Feb-12	21715
15	1-Mar-12	24497
16	1-Apr-12	21720

What are time series data?

Outcome = Deaths per 1000 population



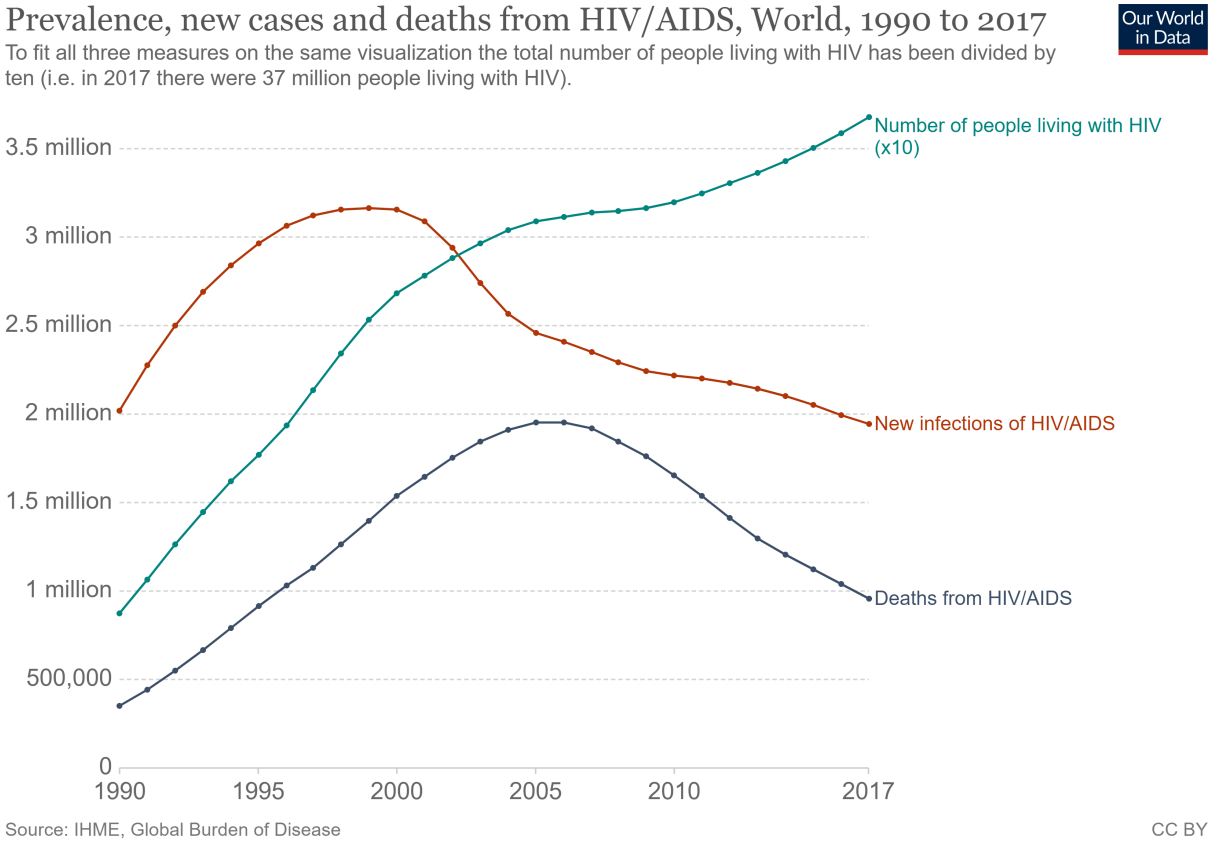
Frequency = Week

UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

Why are time series data useful?

- Understand how health outcomes are changing over time;
- Compare trends in different measures, or in different jurisdictions;
- Compare current trends to historical trends;
- Forecasting of future values;
- Infectious disease surveillance;
- To evaluate population-level health interventions (*which we will cover in the next Chapter*).



Time = Year

Outcome = Count of new HIV cases, deaths, and people living with HIV

UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

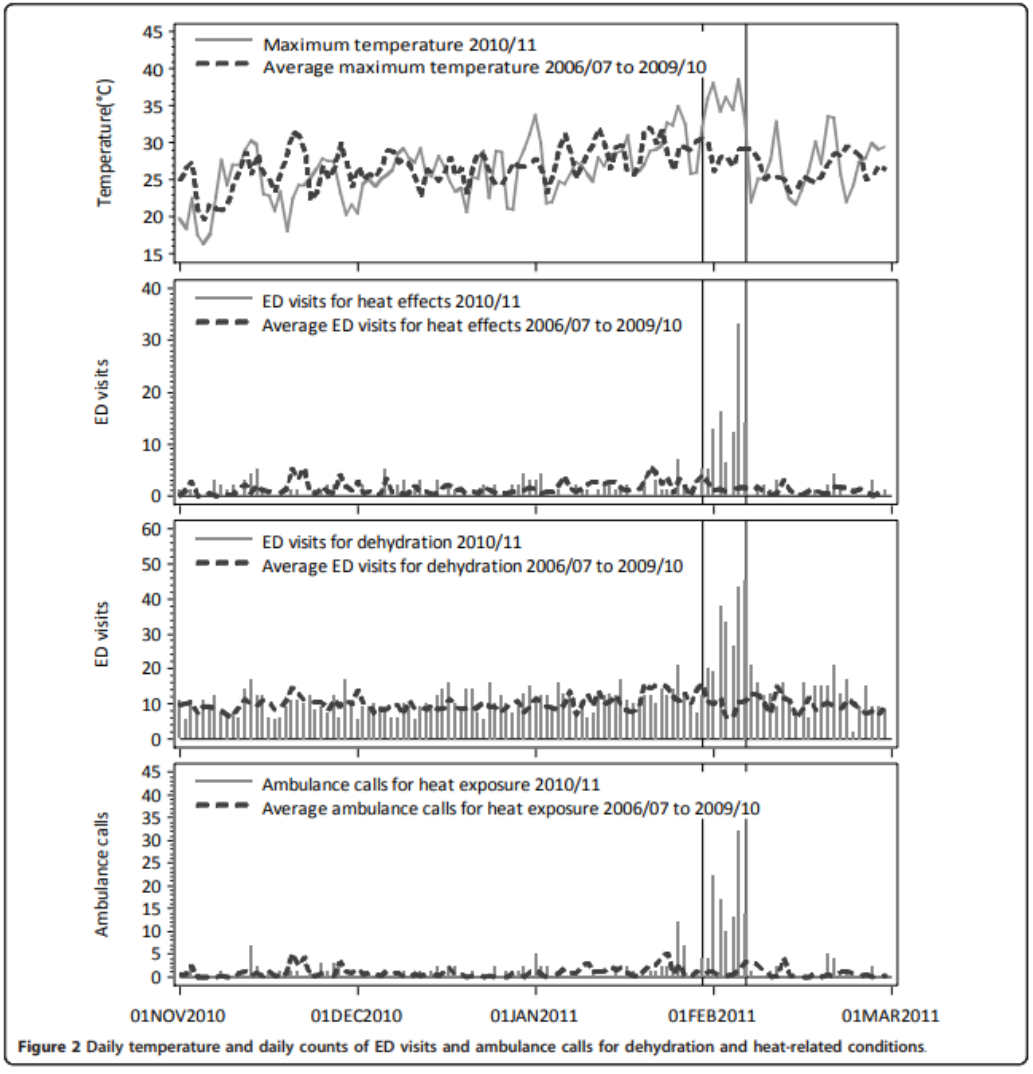


Figure 2 Daily temperature and daily counts of ED visits and ambulance calls for dehydration and heat-related conditions.

Time = Day

Outcome = Maximum and average daily temperature; count and average of ED visits and ambulance calls

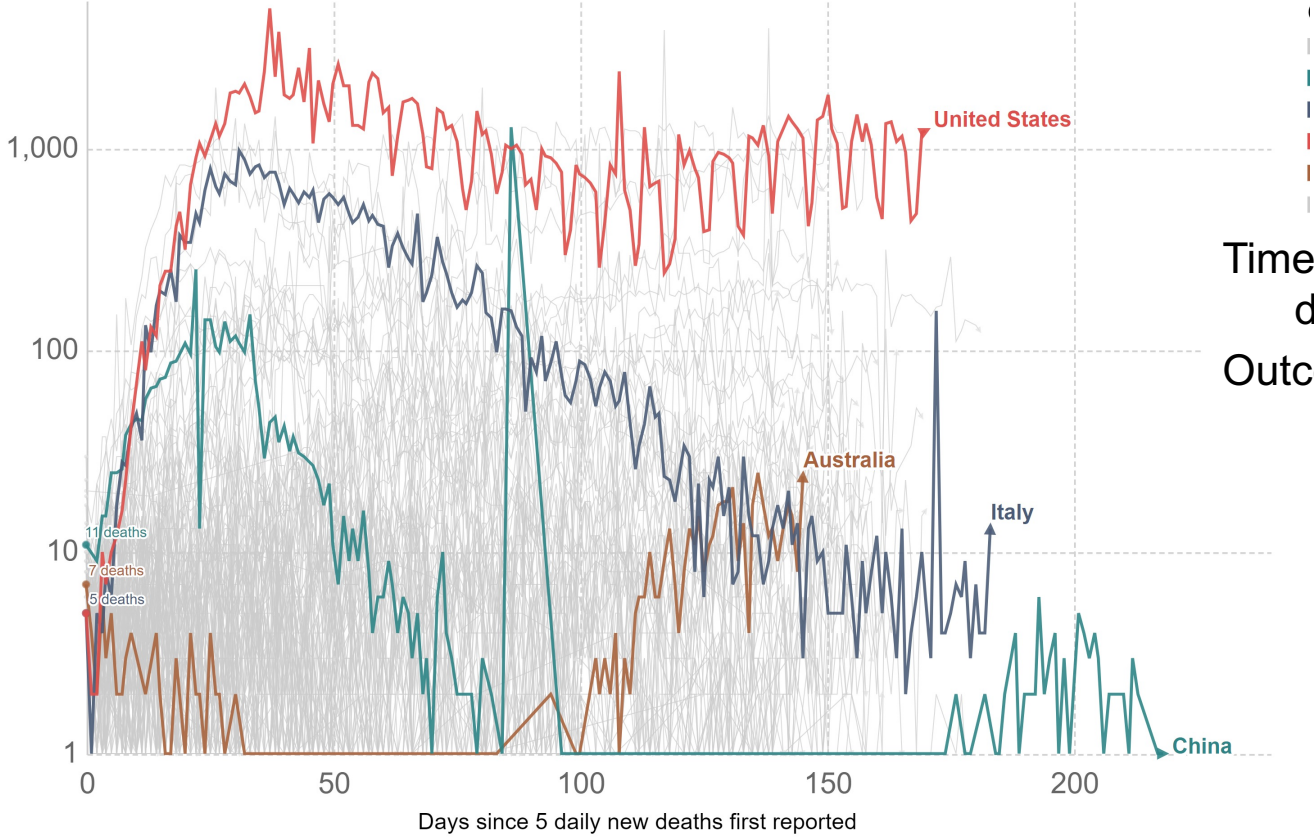
Source: Schaffer et al. *Environ Health*. 2012;11:3.

UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

Daily new confirmed COVID-19 deaths

Limited testing and challenges in the attribution of the cause of death means that the number of confirmed deaths may not be an accurate count of the true number of deaths from COVID-19.



Time = No. days since 5 deaths

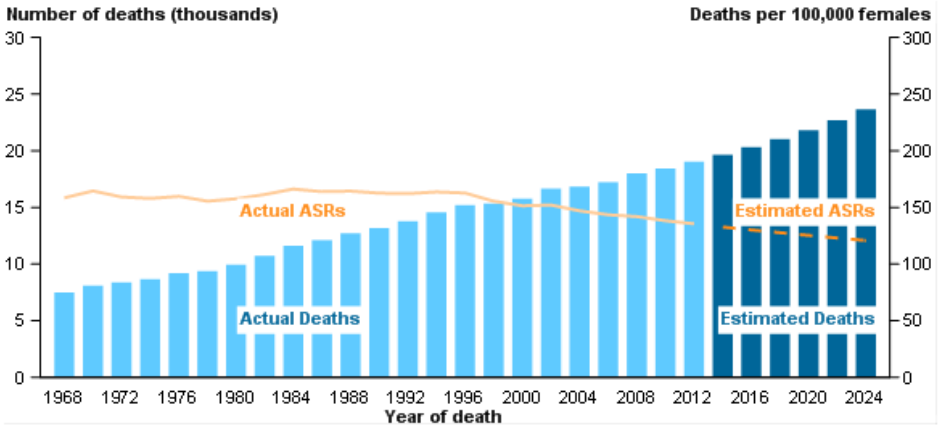
Outcome = Count of deaths

Source: European CDC – Situation Update Worldwide – Last updated 27 August, 10:34 (London time), Our World In Data

UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

Figure 2: Trends in the number and age-standardised rate of deaths due to cancer, trend 1968–2013 and projected 2014 to 2025: females, all cancers combined



Notes

1. All cancers combined includes ICD-10 codes C00–C97, D45, D46, D47.1, D47.3–D47.5.
2. Projected estimates are based on mortality data for all cancers combined between 1995 and 2013, and ABS population projections.
3. Rates are age-standardised to the Australian population as at 30 June 2001, and are expressed per 100,000 females.

Source: AIHW National Mortality Database, Cancer mortality trends and projections: 2014 to 2025 (Data table).

Time = Year

Outcome = Number and age-standardised rate of cancer deaths in females, observed and projected

Working with time series data in R – ts objects

```
data.ts <- ts(data, start=, end=, frequency=)
```

Examples – monthly data:

```
data.ts <- ts(data, start=c(2005,1), end=c(2009,12), frequency=12)

data.ts <- ts(data[,2], start=c(2005,1), end=c(2009,12), frequency=12)
```

Example – yearly data:

```
data.ts <- ts(data, start=2005, end=2009, frequency=1)
```

Example – quarterly data

```
data.ts <- ts(data, start=c(2005,1), end=c(2009,4), frequency=4)
```

Working with time series data in R – ts objects

Time component is held in an "index", that indicates the relative position of each observation in the ts object

To access the index use:

```
time(data.ts)
```

OR

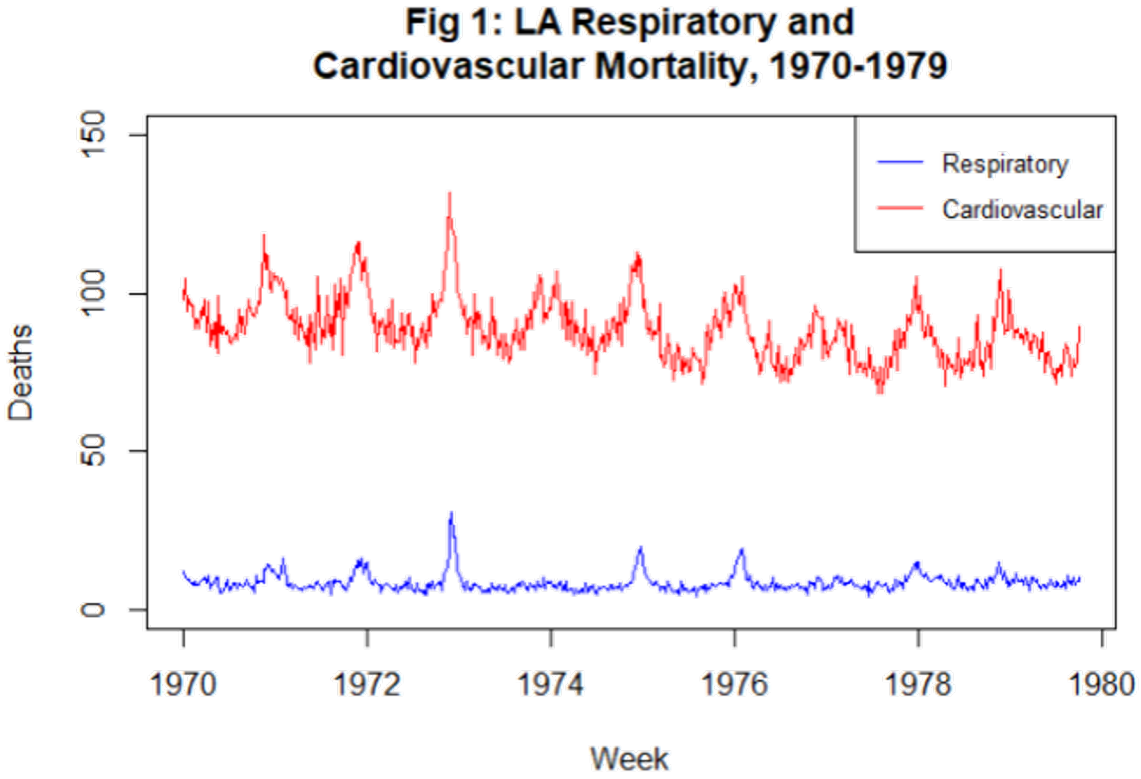
```
as.yearmon(time(data.ts))
```

```
> time(x1)
          Jan      Feb      Mar      Apr      May      Jun
2005
2006 2006.000 2006.083 2006.167 2006.250 2006.333 2006.417
2007 2007.000 2007.083 2007.167 2007.250 2007.333 2007.417
2008 2008.000 2008.083 2008.167 2008.250 2008.333 2008.417
2009 2009.000 2009.083 2009.167 2009.250 2009.333 2009.417
          Jul      Aug      Sep      Oct      Nov      Dec
2005 2005.500 2005.583 2005.667 2005.750 2005.833 2005.917
2006 2006.500 2006.583 2006.667 2006.750 2006.833 2006.917
2007 2007.500 2007.583 2007.667 2007.750 2007.833 2007.917
2008 2008.500 2008.583 2008.667 2008.750 2008.833 2008.917
2009

> as.yearmon(time(x1))
 [1] "Jul 2005" "Aug 2005" "Sep 2005" "Oct 2005" "Nov 2005"
 [6] "Dec 2005" "Jan 2006" "Feb 2006" "Mar 2006" "Apr 2006"
[11] "May 2006" "Jun 2006" "Jul 2006" "Aug 2006" "Sep 2006"
[16] "Oct 2006" "Nov 2006" "Dec 2006" "Jan 2007" "Feb 2007"
[21] "Mar 2007" "Apr 2007" "May 2007" "Jun 2007" "Jul 2007"
[26] "Aug 2007" "Sep 2007" "Oct 2007" "Nov 2007" "Dec 2007"
[31] "Jan 2008" "Feb 2008" "Mar 2008" "Apr 2008" "May 2008"
[36] "Jun 2008" "Jul 2008" "Aug 2008" "Sep 2008" "Oct 2008"
[41] "Nov 2008" "Dec 2008" "Jan 2009" "Feb 2009" "Mar 2009"
[46] "Apr 2009" "May 2009" "Jun 2009"
```

Visualisation

Always start by visualising your data!!



What makes a good figure?

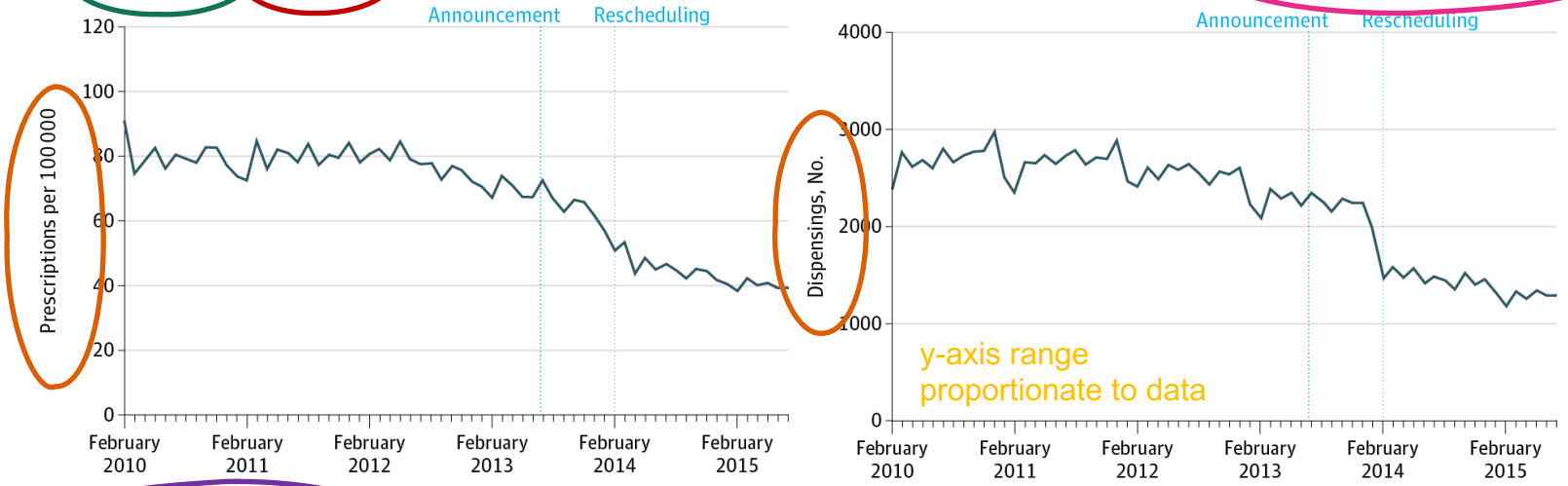
Figure label

Clear indication of frequency

Population named

Figure 1. Monthly Time Series of Alprazolam Prescriptions per 100 000 Australian Population

Axis clearly labelled, units clear



Legend. The rescheduling was announced on June 28, 2013, whereas the rescheduling took effect on February 1, 2014.

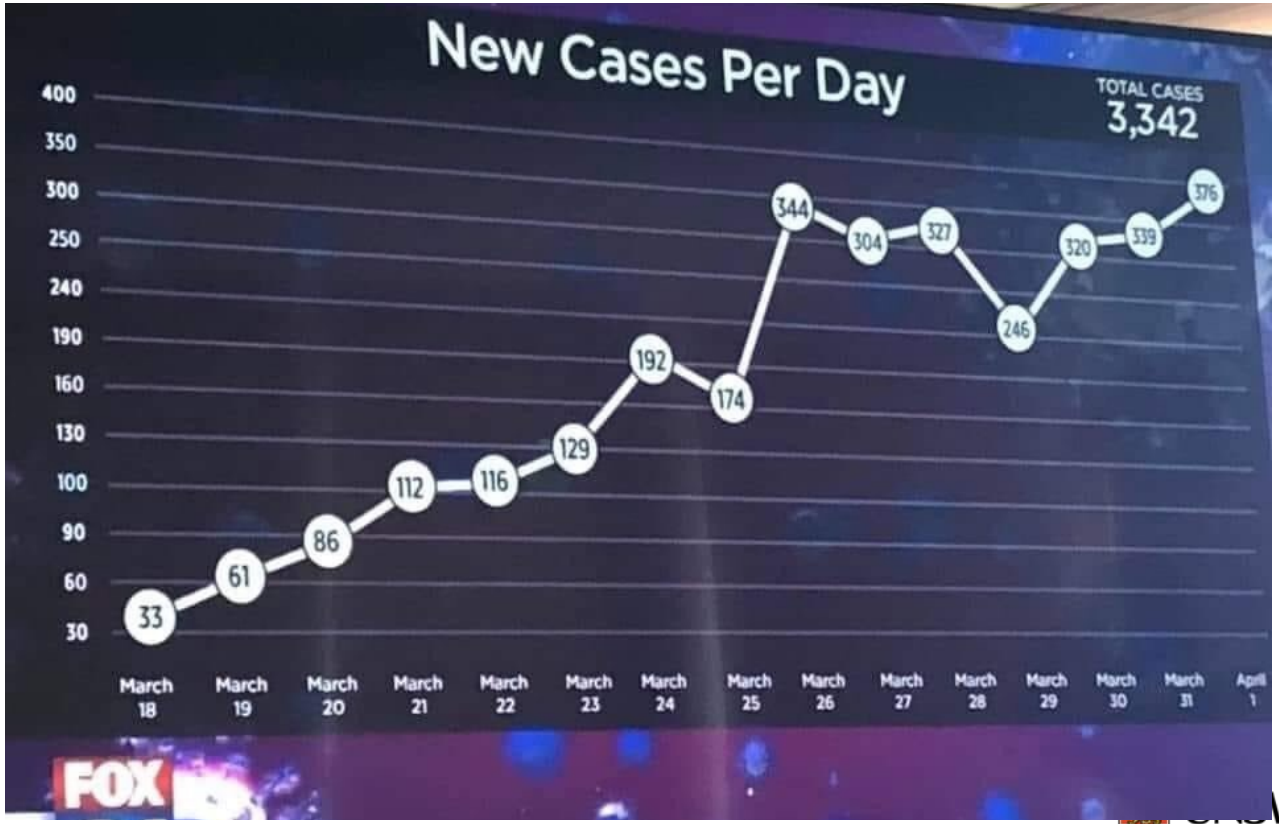
Extra information in legend

Schaffer et al. 2016. *JAMA Intern Med*

UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

What makes a bad figure?



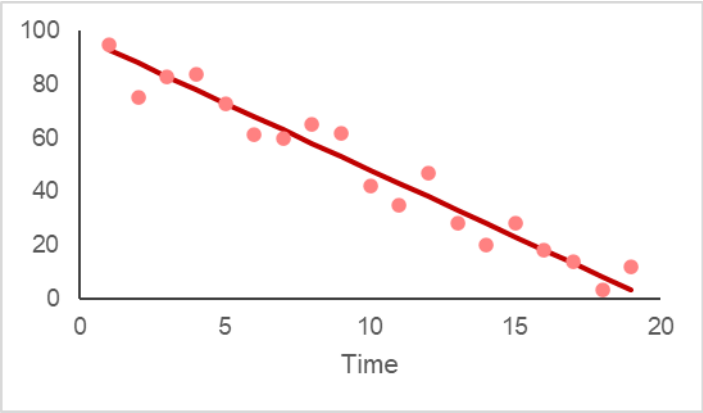
Time series decomposition

1. Trend
2. Seasonality
3. Error (random variation)

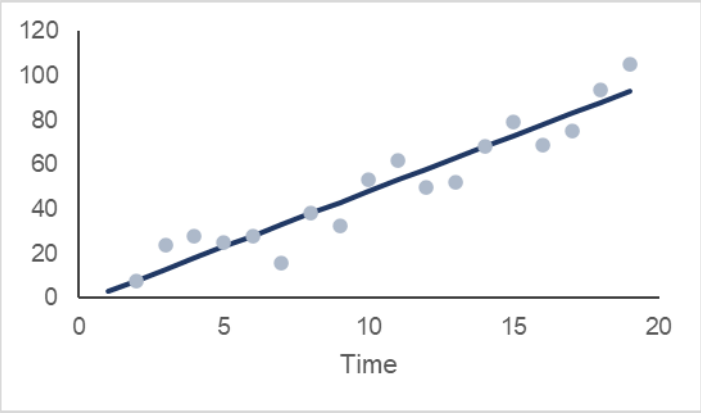
What is a trend?

Trend: a long-term increase or decrease in the data—it can be linear (best represented by a straight line), or non-linear (e.g. exponential). Some time series have no trend.

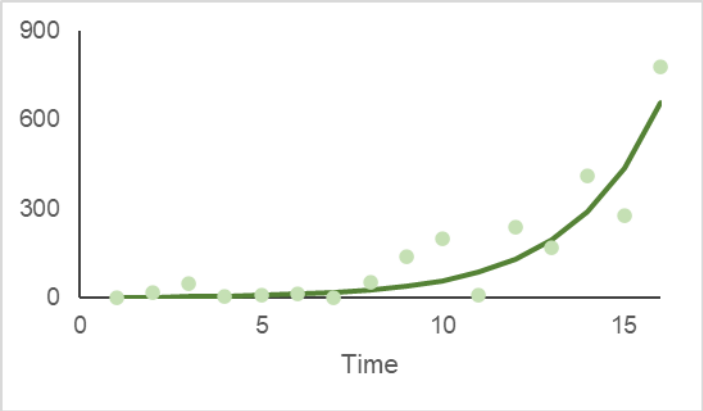
Downward linear trend



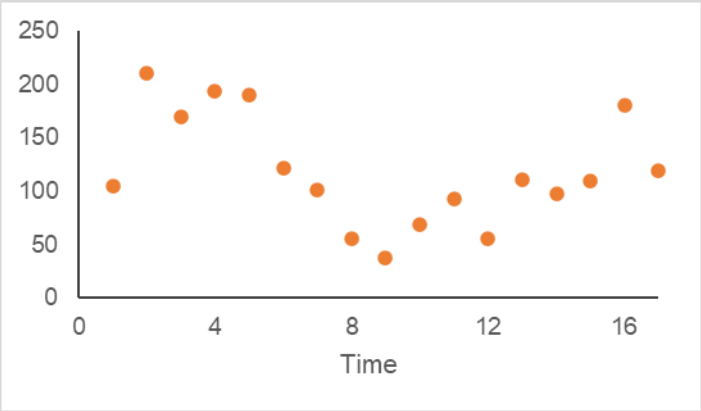
Upward linear trend



Exponential trend



???



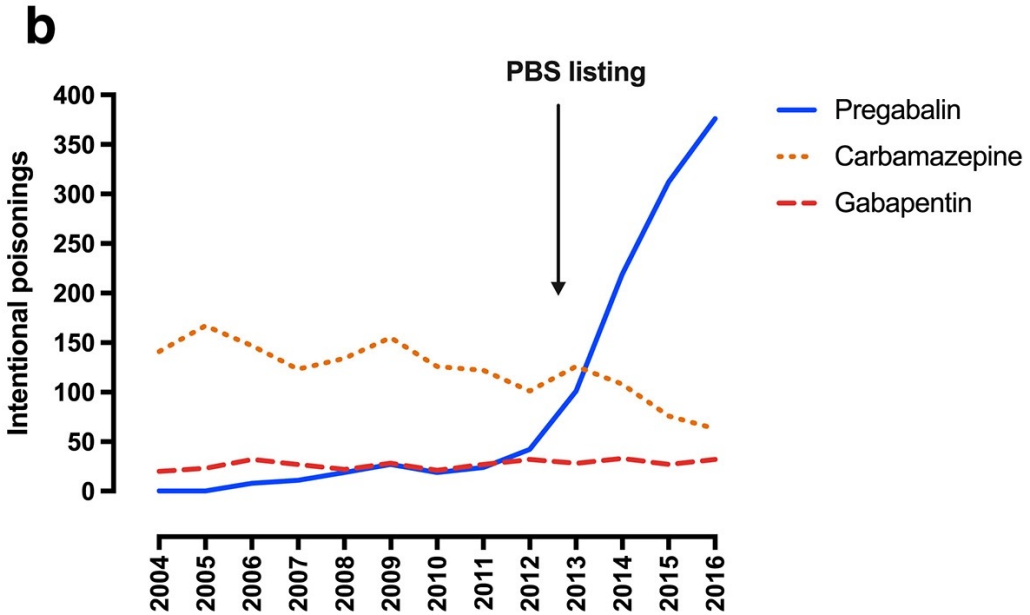


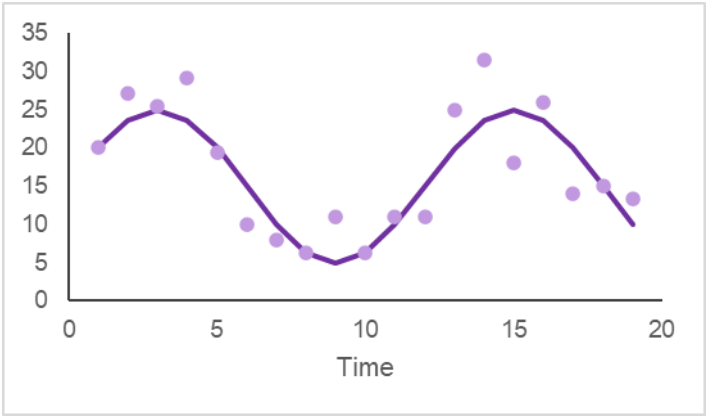
Figure 1 (b) intentional poisonings with pregabalin, gabapentin and carbamazepine reported to the NSWPIC, 2004–2016. Arrow shows PBS listing, 2013. PBS = Pharmaceutical Benefits Scheme; NSWPIC = New South Wales Poisons Information Centre.

Source: Cairns et al. *Addiction.* 2019;144:1026.

What is seasonality?

Seasonality: the outcome is correlated with calendar time, such as time of year or day of the week

- It is always of a fixed frequency
- It can be due to natural causes or business/administrative processes
- Seasonality will depend on the time unit or frequency of your series (e.g. daily, weekly)
- Yearly time series do not have seasonality! (at least not health data)

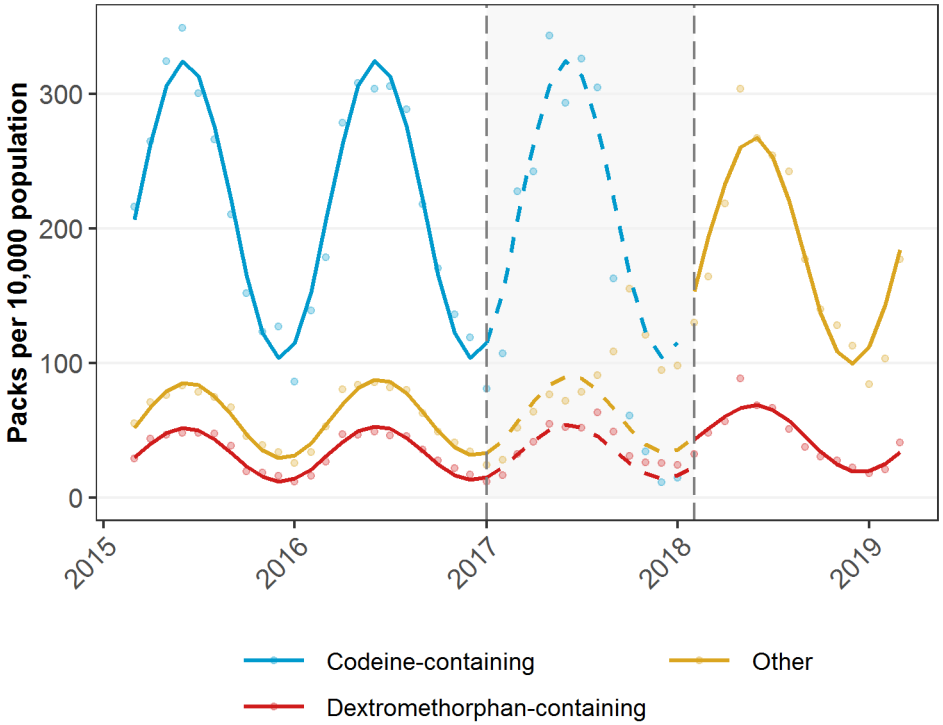


Example: Monthly sales of cold and flu products

When are sales highest? lowest?

Why?

Example: Monthly sales of cold and flu products



When are sales highest? lowest?

Why?

Source: Schaffer et al. *Med J Aust.* 2020 Apr;212(7):321-327

UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

What is error/random variation? What are outliers?

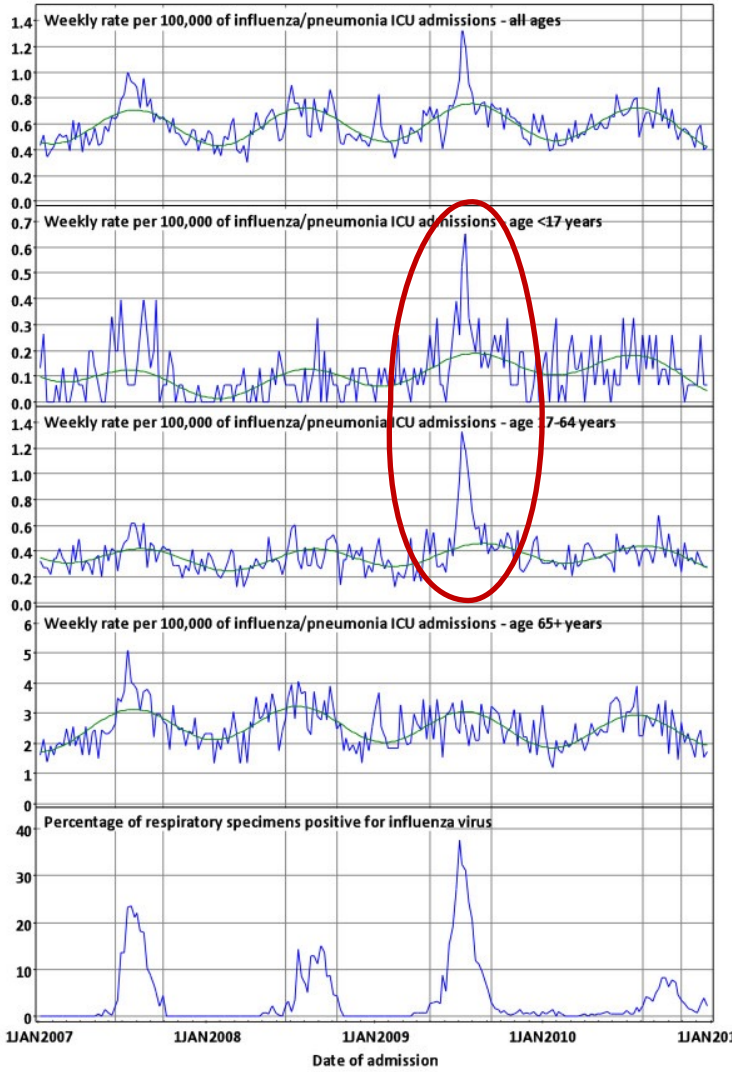
Error (aka random variation): it is the variation in the series not explained by trend or seasonality.

Outliers: extreme values in your series—they may occur randomly, or due to an underlying process.

How to deal with extreme values/outliers?? – it depends!

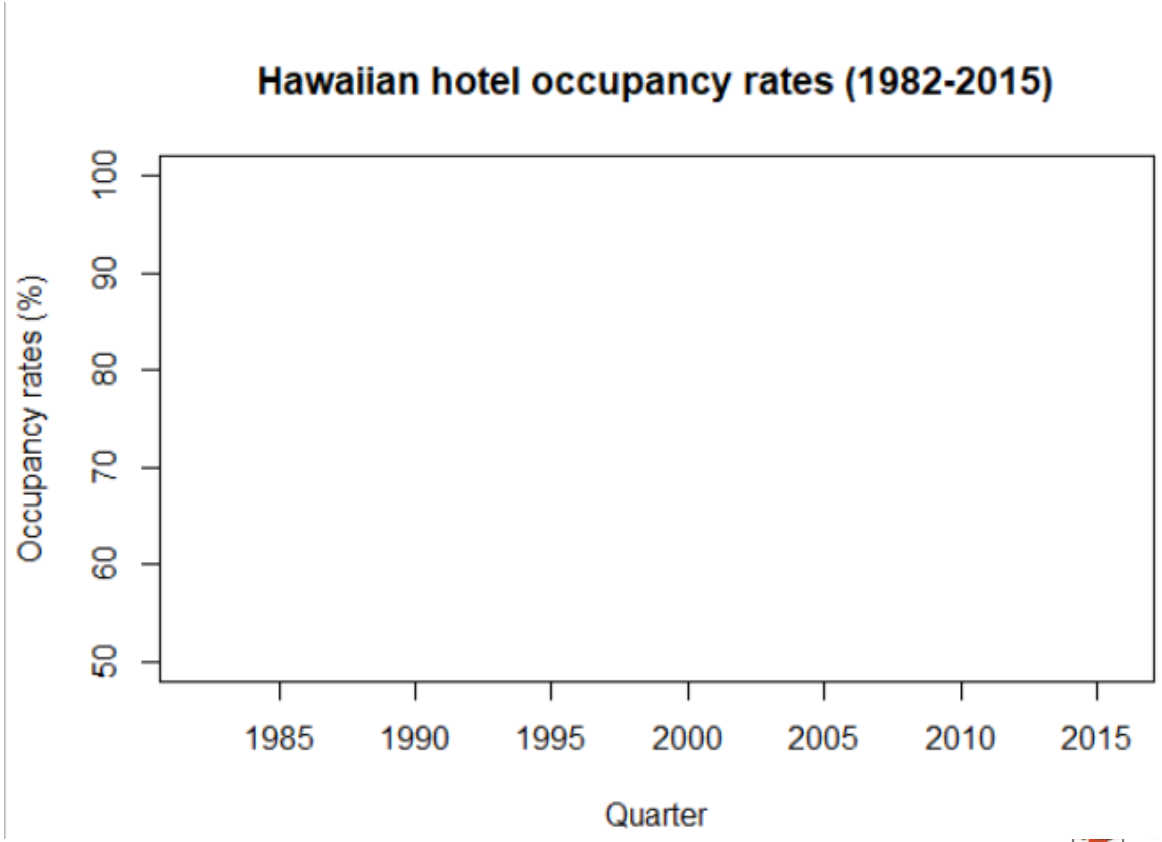
Example: ICU admissions for influenza/pneumonia

ICU admissions were very high in winter 2009, especially in people <65 years—this was due to influenza A(H1N1)pdm09 which was circulating that year

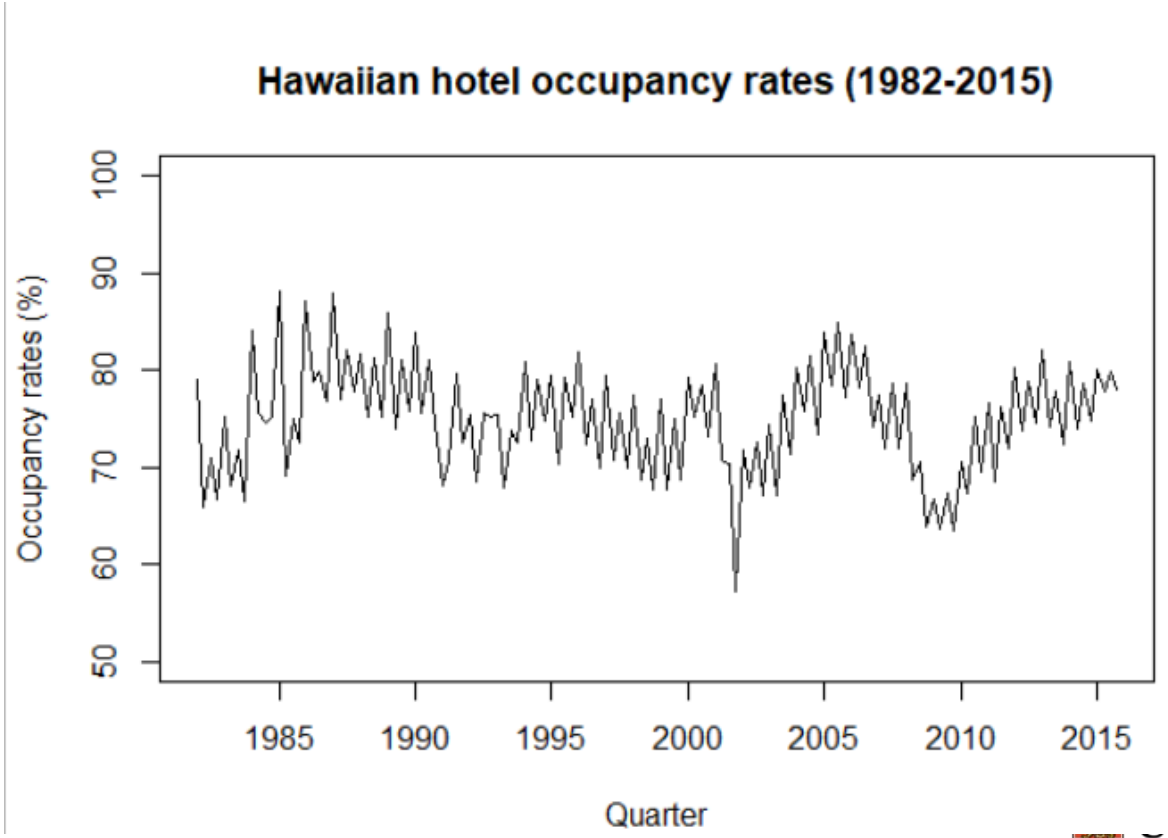


Source: Schaffer et al. *BMC Public Health* 2012 Oct 12;12:869.

Example: Time series decomposition



Example: Time series decomposition

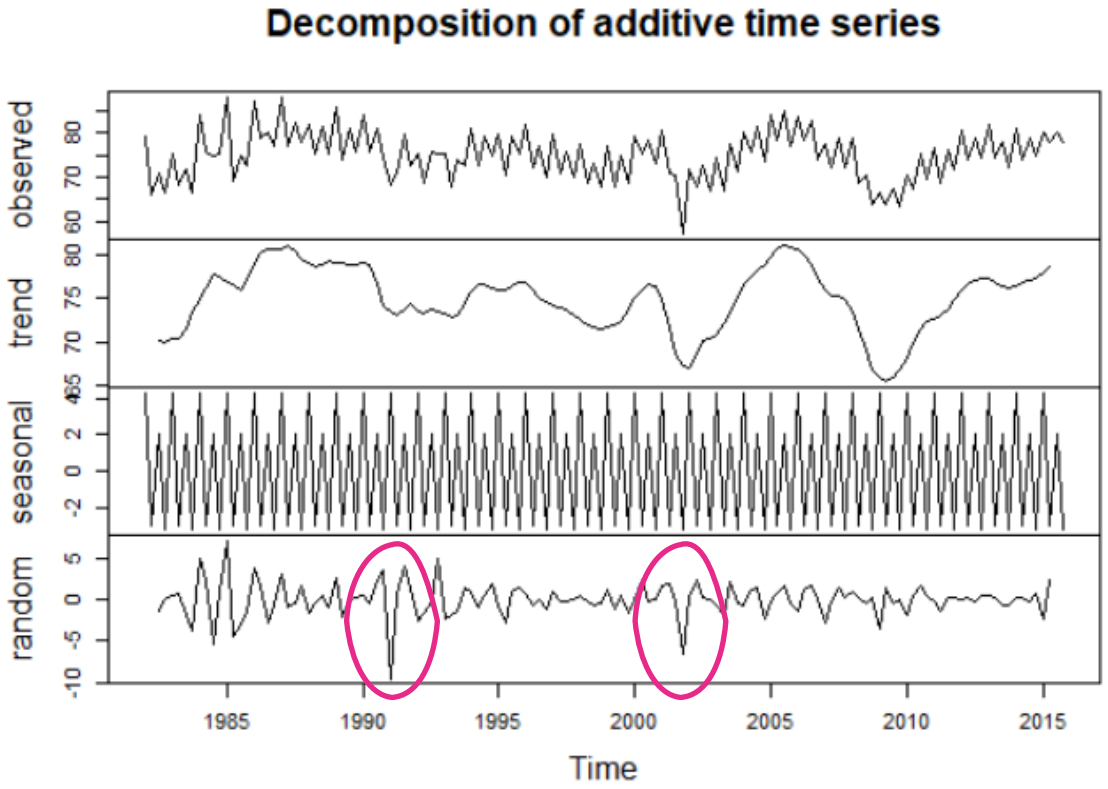


Example: Time series decomposition

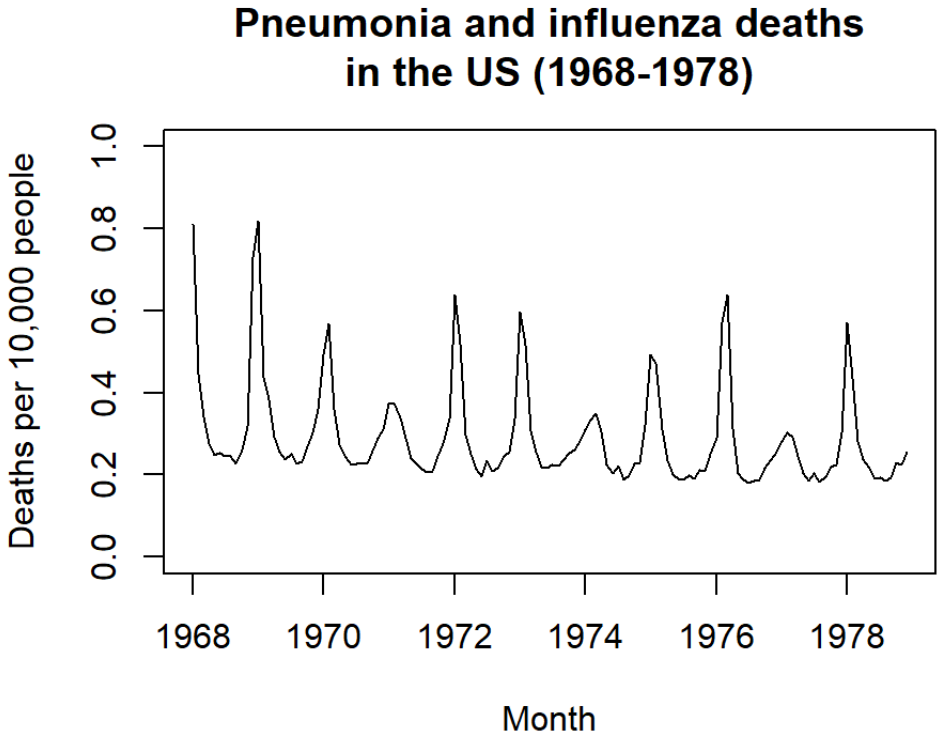
```
decompose(data=,
    type=c("additive",
    "multiplicative"))
```

Decomposition plot of Hawaiian hotel occupancy data:

```
plot(decompose(hor))
```



A better way to visualise seasonal effects



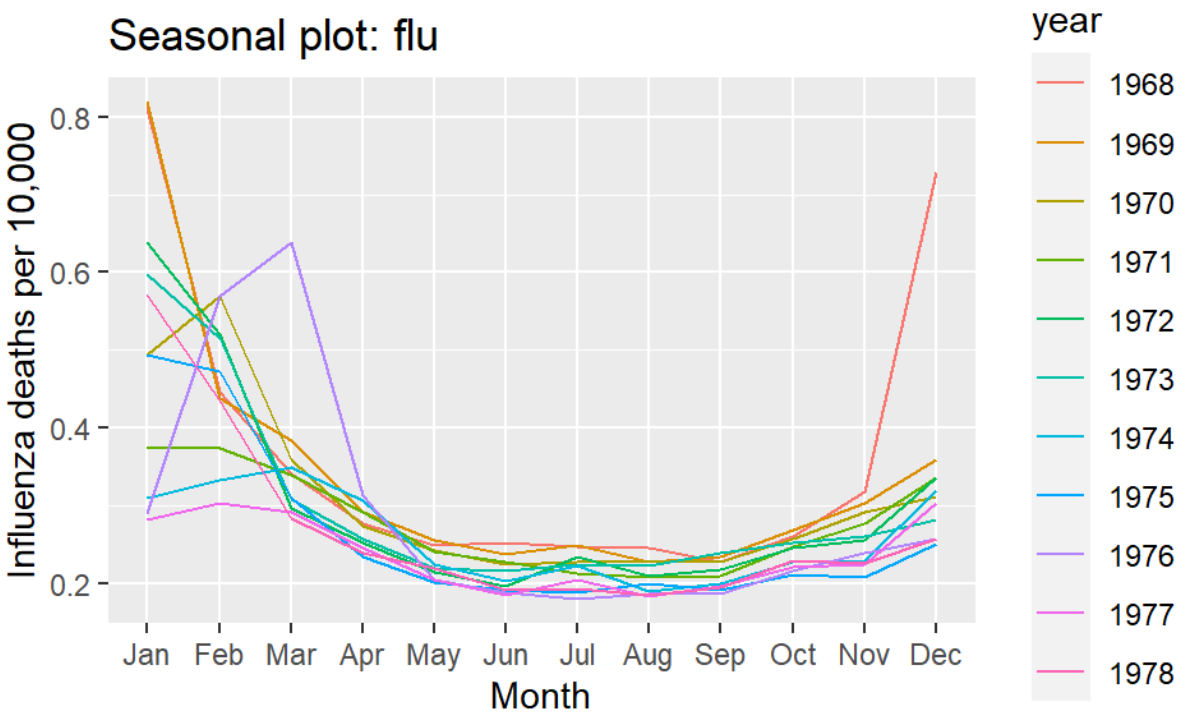
Seasonal plot of influenza mortality rate in the US:

```
ggseasonplot()
```

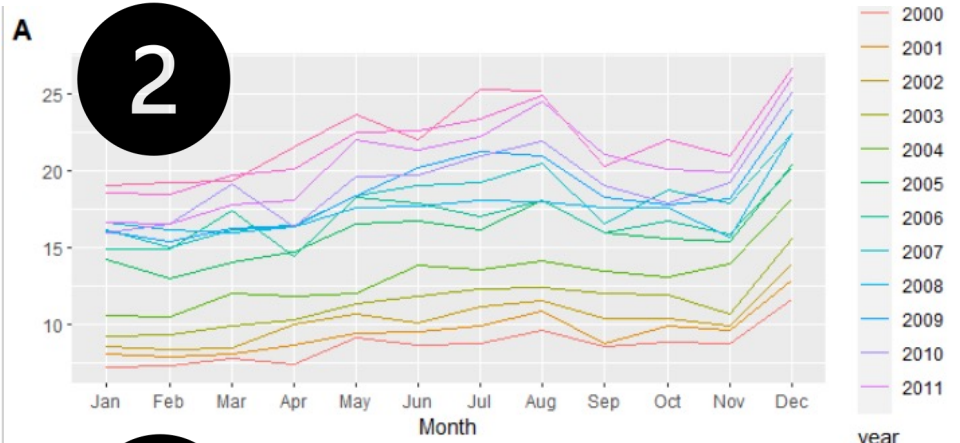
A better way to visualise seasonal effects

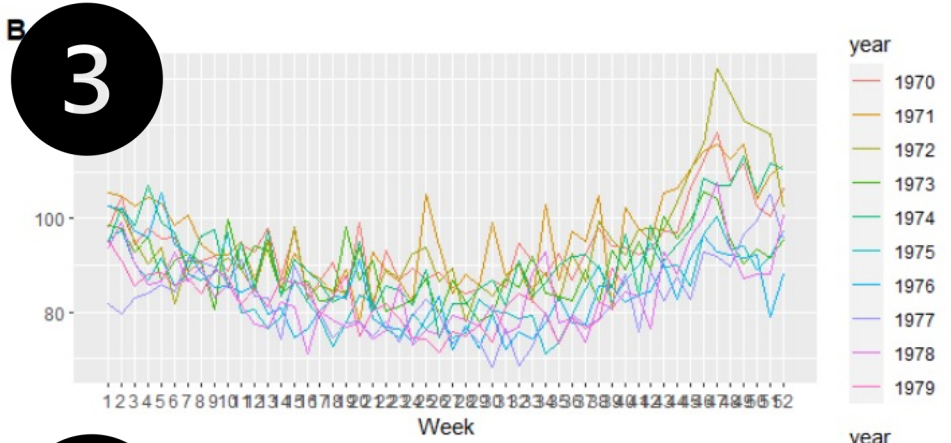
```
ggseasonplot(flu)
```

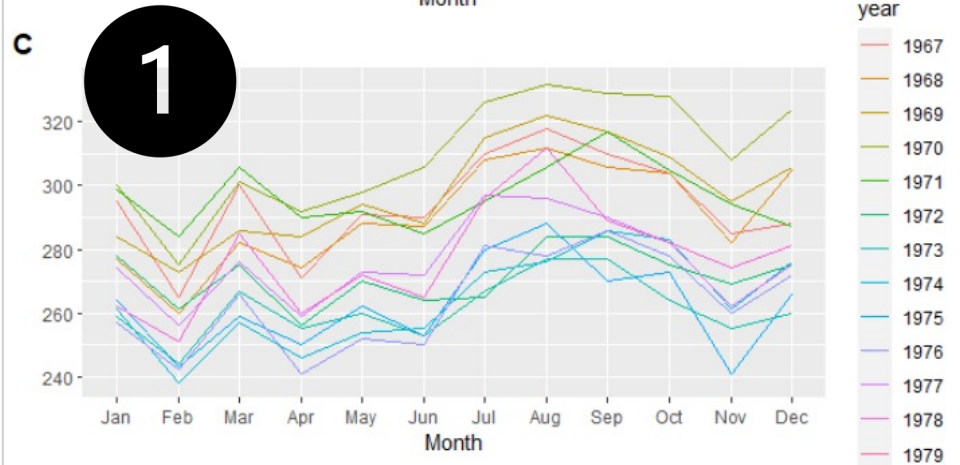
What does this plot tell you?

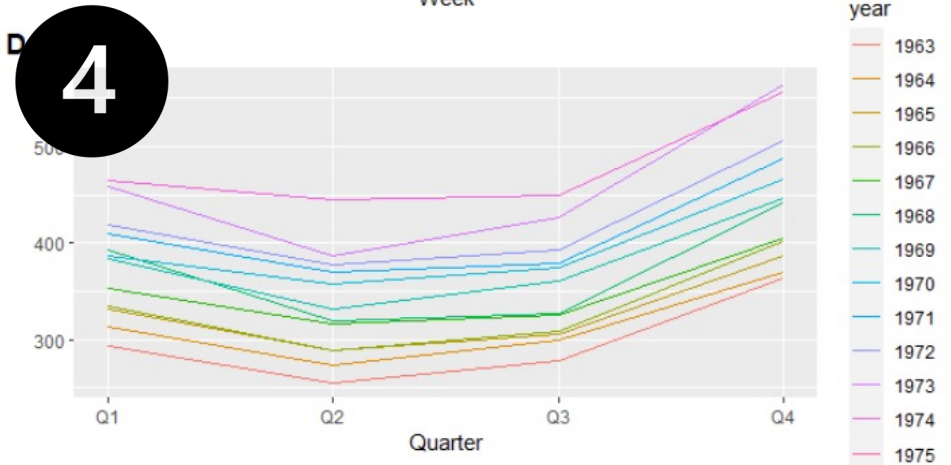


Which of the following is: 1. Births in the US? 2. Debit card spending in Iceland? 3. Cardiovascular mortality in LA? 4. Beer production in Australia?









What time/season-related factors exist for the following measures:

1. Dispensing of chronic medicines (e.g. statins, antihypertensives, diabetes medicines)?
2. Dispensing of antibiotics?
3. GP visits?
4. Emergency department visits?

Statistical properties of time series

1. Stationarity
2. Autocorrelation
3. Seasonality

What is stationarity?

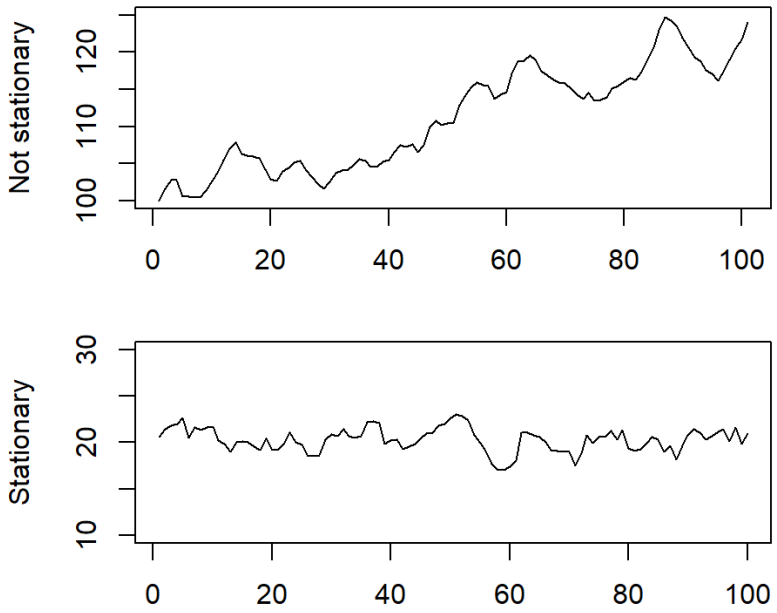
In simple terms, a stationary series has no trend, and no autocorrelation—therefore, it has a constant mean and constant variance.

This means it is easier to predict!

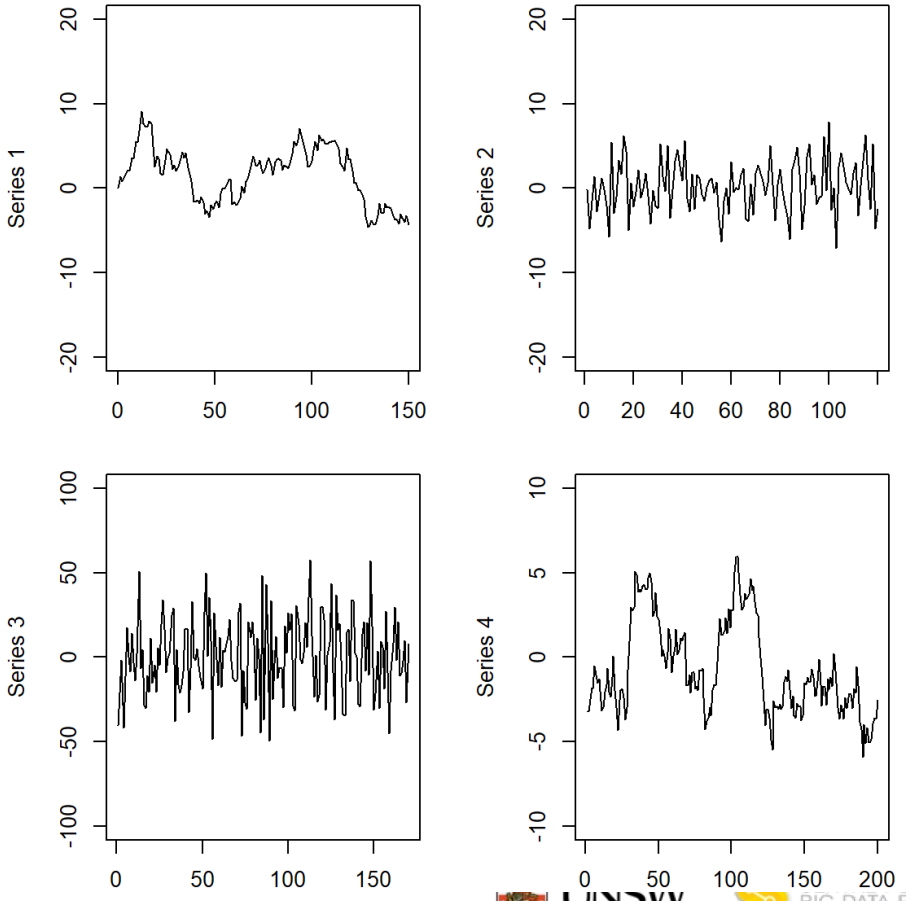
Stationarity

Definition of weak stationarity:

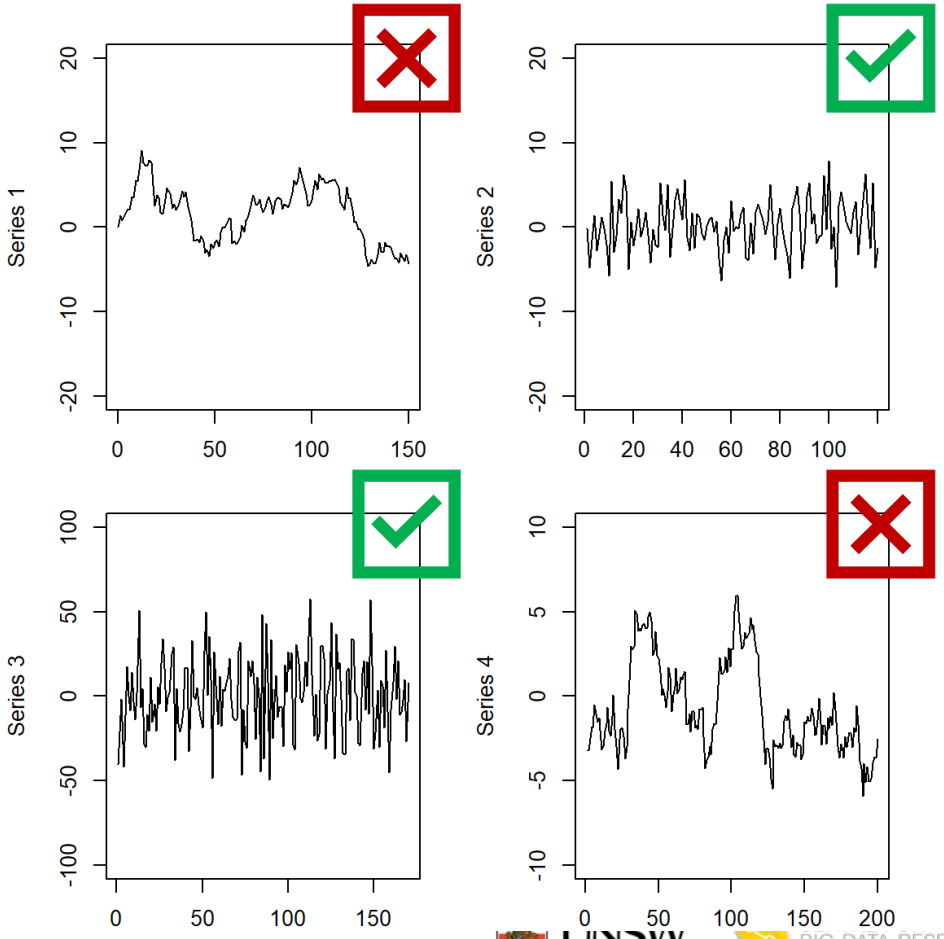
1. the mean of Y_t is the same for all *t*;
2. the variance of Y_t is the same for all *t*;
3. the covariance of Y_t and Y_{t-m} is the same for all *t* and depends only on *m* (the distance between values, or the lag).



Which of these series appear stationary?



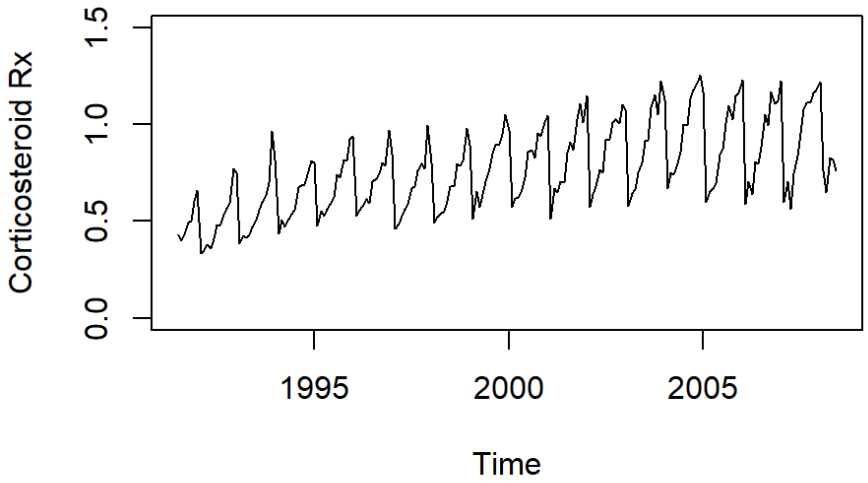
Which of these series appear stationary?



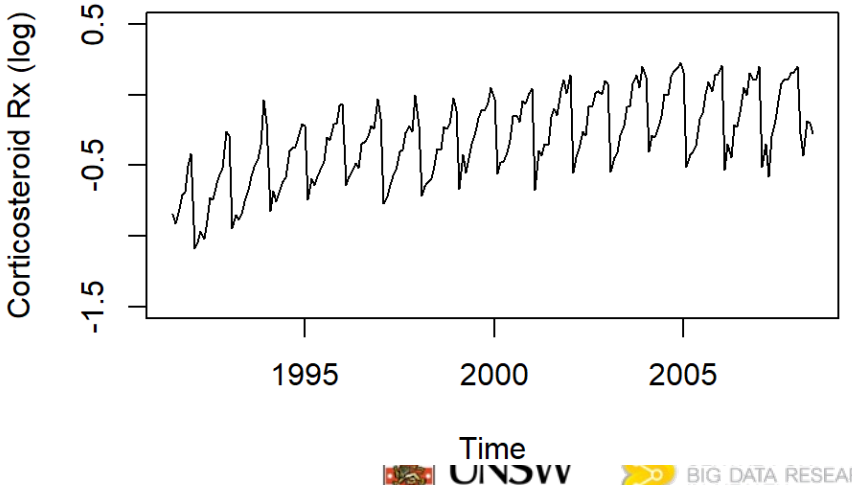
How to deal with non-stationarity

First: does your series have constant variance? → If not, try a log (natural) transformation

`plot(h02)`



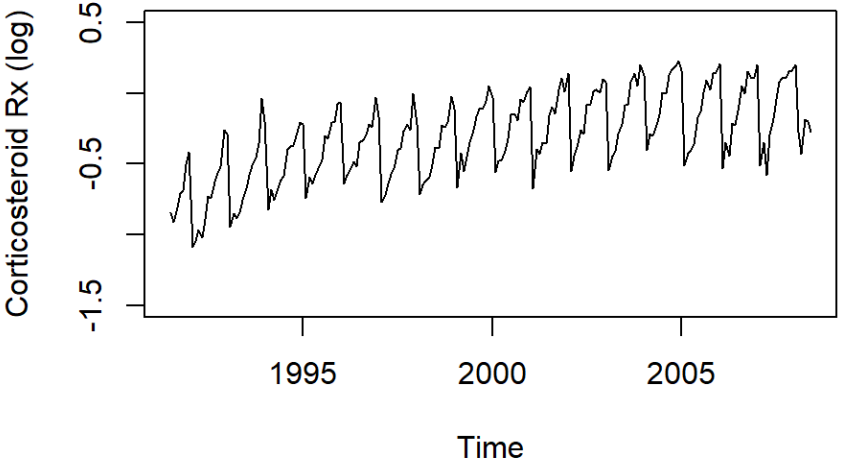
`plot(log(h02))`



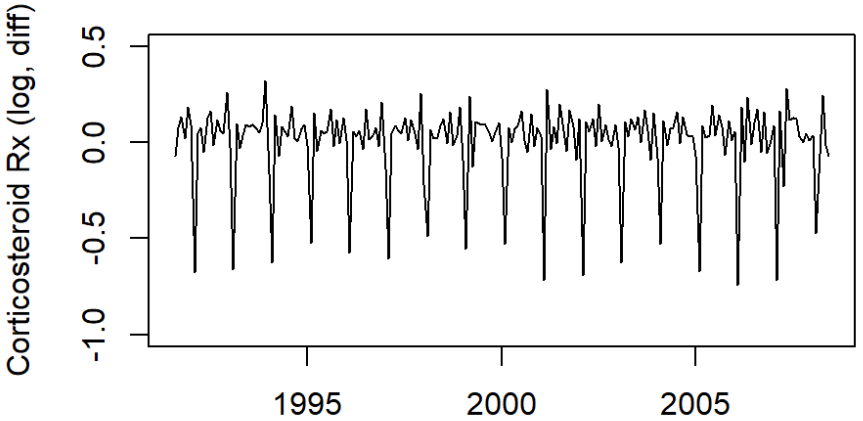
How to deal with non-stationarity

Second: does your series have a trend? → If so, try taking the first difference of your series ($Y_t - Y_{t-1}$)

`plot(log(h02))`



`plot(diff(log(h02)))`



***If needed, always log transform your data before taking the difference!!

What is autocorrelation?

Autocorrelation refers to correlation (or dependence) between observations in your time series (Y_t) and its previous value(s).

An assumption of linear regression is that your residuals are independent (not correlated)!

If you don't adjust for autocorrelation, you may get biased results or incorrect standard errors.

What do we mean when we talk about lag??

Time	Observation		Lag (in reference to Y_t)
…	…	…	…
8	23467	Y_{t-4}	4
9	33896	Y_{t-3}	3
10	31053	Y_{t-2}	2
11	28482	Y_{t-1}	1 (first-order lag)
12	38900	Y_t	0

If Y_t is correlated with Y_{t-1}, this is called first-order autocorrelation

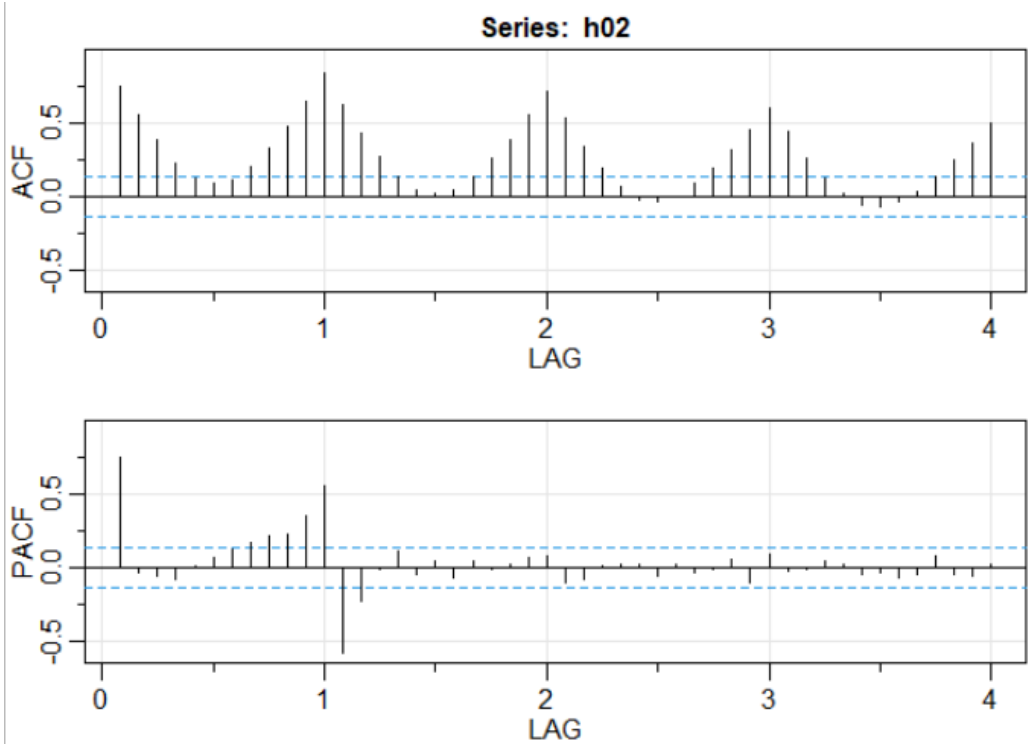
But Y_t could also be correlated with Y_{t-2}, Y_{t-3}, Y_{t-4}.

Autocorrelation function (ACF) and partial ACF plots

```
acf2(data=, lag=)
```

ACF/PACF plots for corticosteroid prescriptions

```
acf2(h02)
```



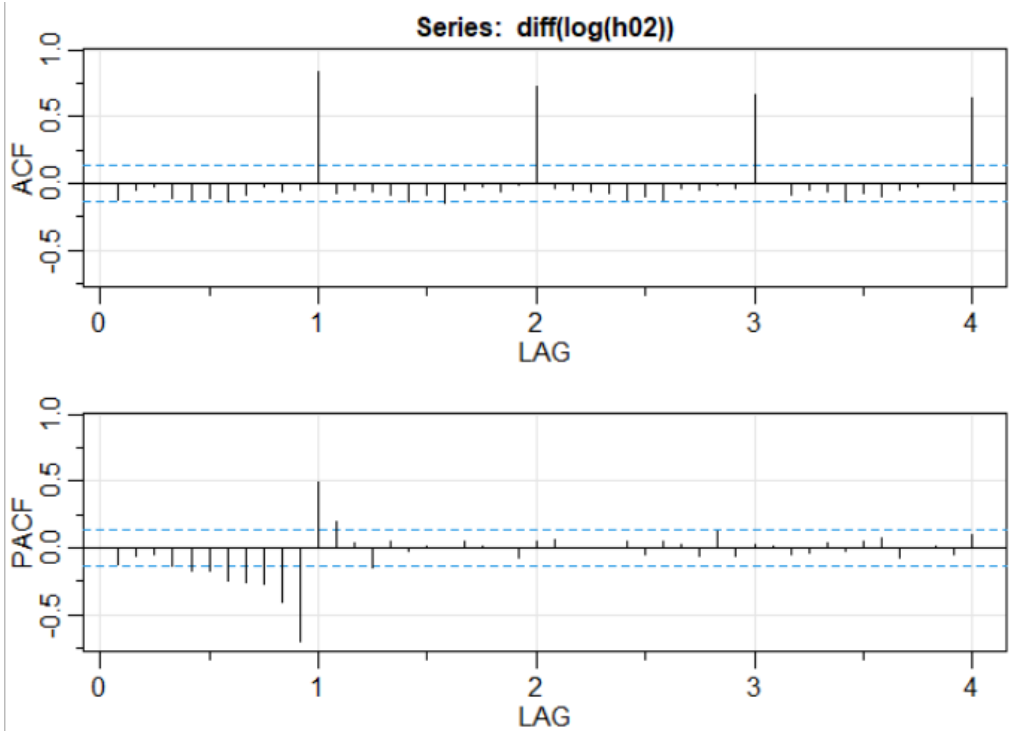
UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

Autocorrelation function (ACF) and partial ACF plots

You should always check the ACF/PACF on your stationary series!

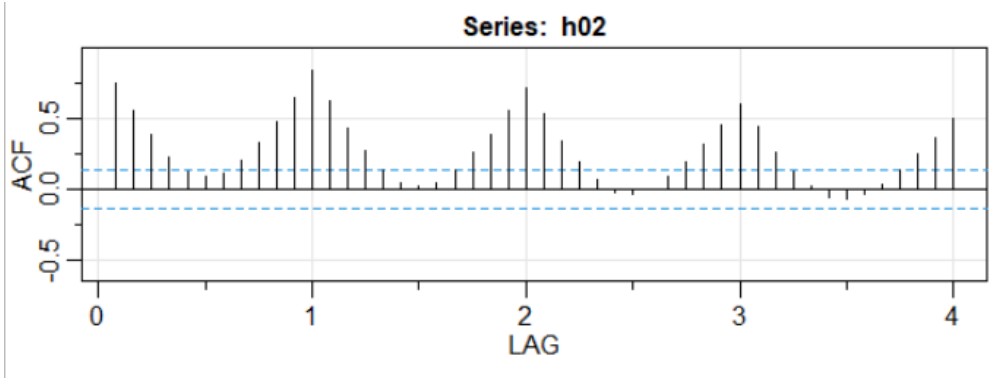
```
acf2(diff(log(h02)))
```



What is seasonality?

Seasonality is a type of autocorrelation—observations at a given time point are correlated with previous observations at the same time of year.

For instance, with monthly data, an observation in July 2018 (Y_t) would be correlated with the observation at lag 12 in July 2017 (Y_{t-12}).



Seasonal autocorrelation

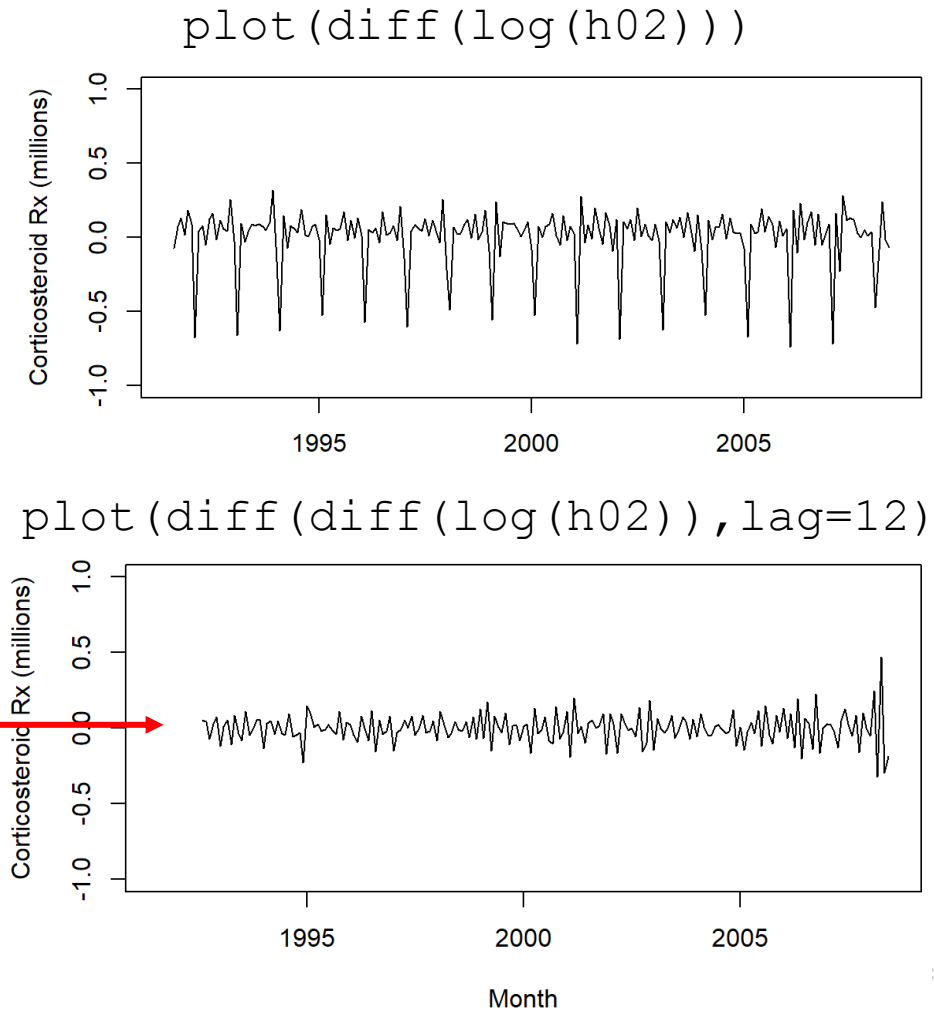
Time	Observation		Lag (in reference to Y_t)
1	30314	Y_{t-12}	12
…	…	…	…
8	23467	Y_{t-4}	4
9	33896	Y_{t-3}	3
10	31053	Y_{t-2}	2
11	28482	Y_{t-1}	1 (first-order lag)
12	38900	Y_t	0

If your data are monthly and exhibit seasonality, then Y_t is correlated with Y_{t-12}.

Seasonality

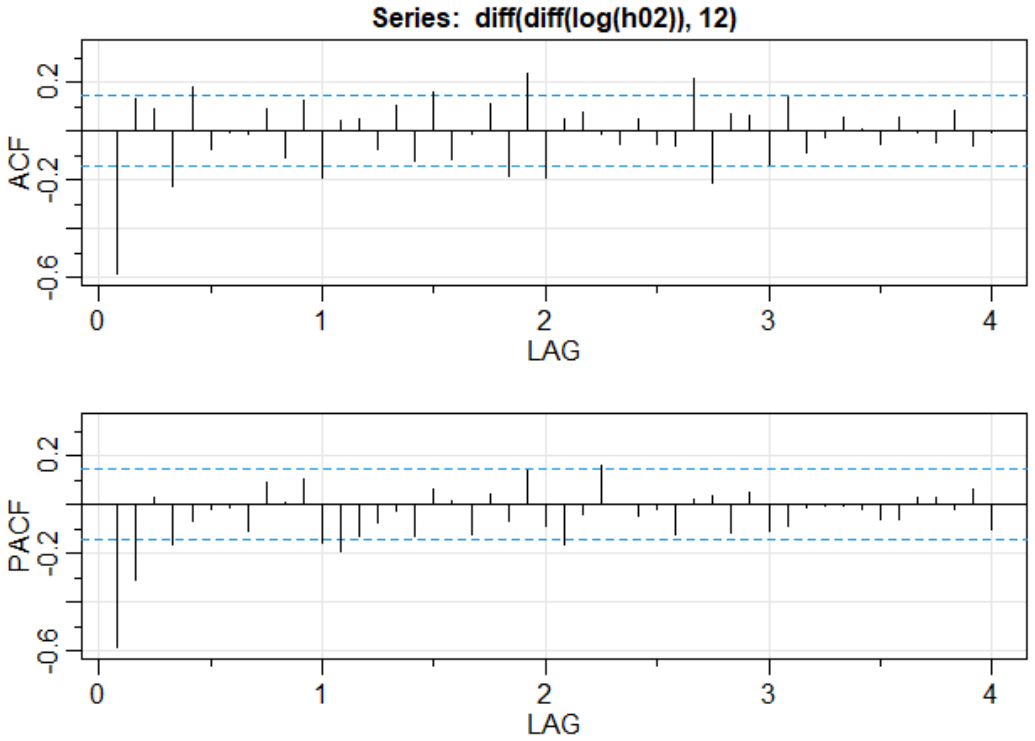
Seasonality can be dealt with by taking the seasonal difference, e.g. for monthly data: $Y_t - Y_{t-12}$.

Your series is now stationary!



Seasonality

```
acf2(diff(diff(log(h02)),
   lag=12)
```

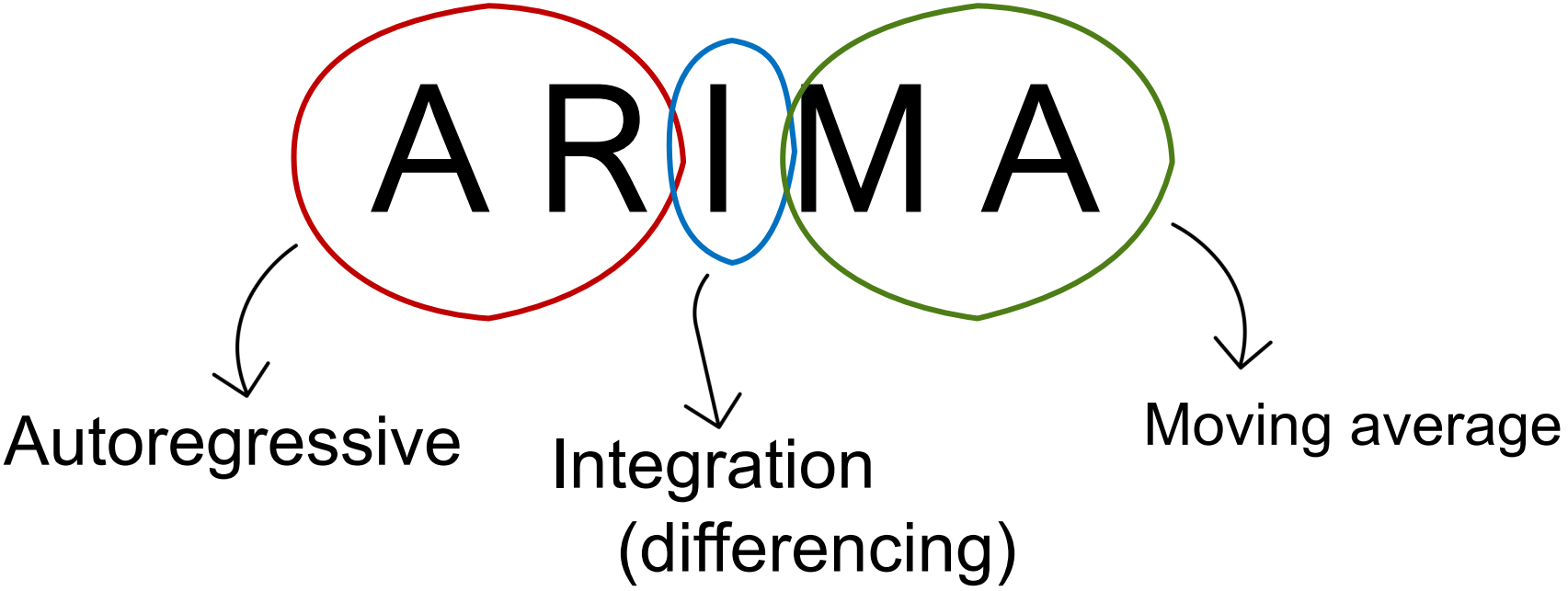


Time series models

Depending on your data, you may be able to use ordinary regression models to analyse your time series—we will cover this next Chapter.

An alternative is **A**uto**r**egressive **I**ntegrated **M**oving **A**verage (ARIMA) models, which can control for non-stationarity, autocorrelation and seasonality.

What is an ARIMA model?



Autoregressive component

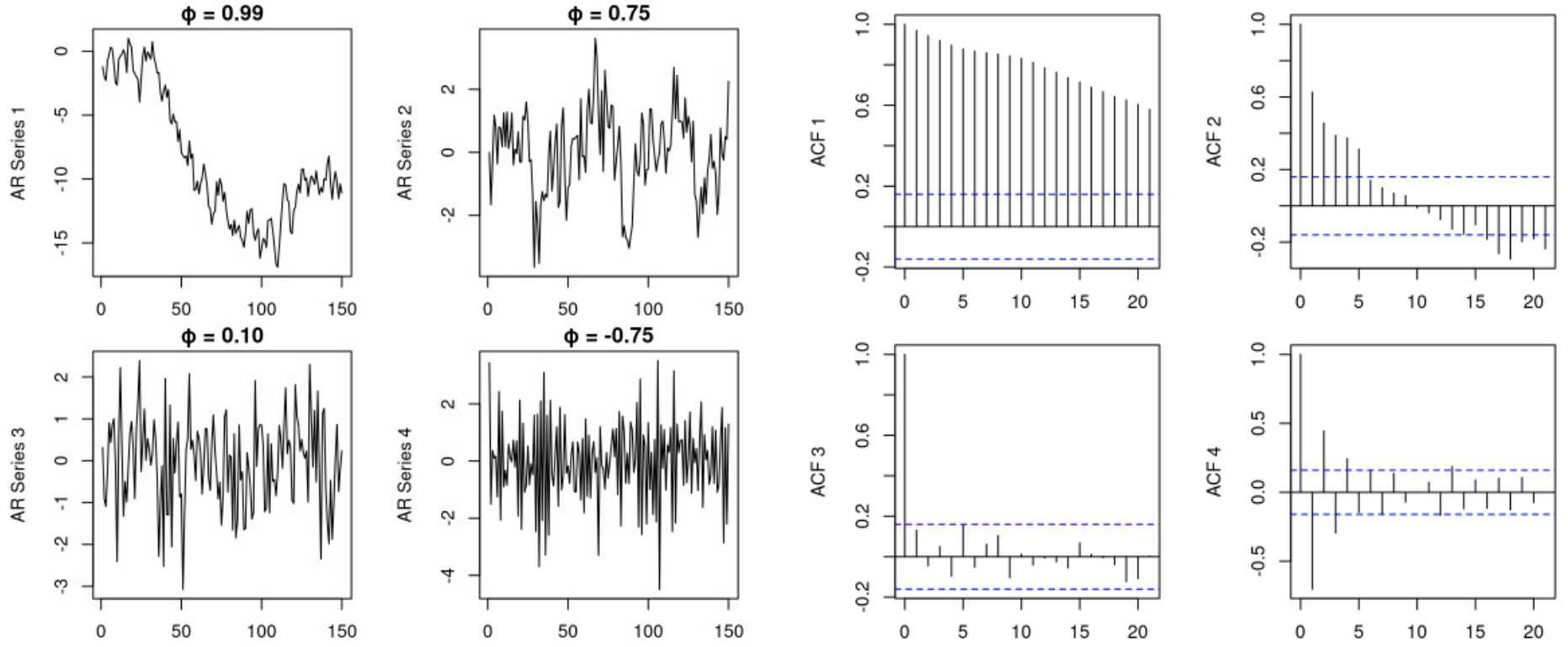
In an autoregressive (AR) model, Y_t is predicted by one or more previous values of Y_t and helps adjust for autocorrelation.

p = order of the AR model

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

ϕ = value of the autocorrelation

Autoregressive component



Integration component

Here, integration means *differencing.*

d is the order of differencing required to generate a stationary series (eliminate trend). If your data have no trend, then $d = 0$.

If your data has a trend, then usually $d = 1$. d is rarely greater than 1.

Note: with ARIMA models, the differencing is done by the ARIMA function, no need to difference the data yourself.

Moving average component

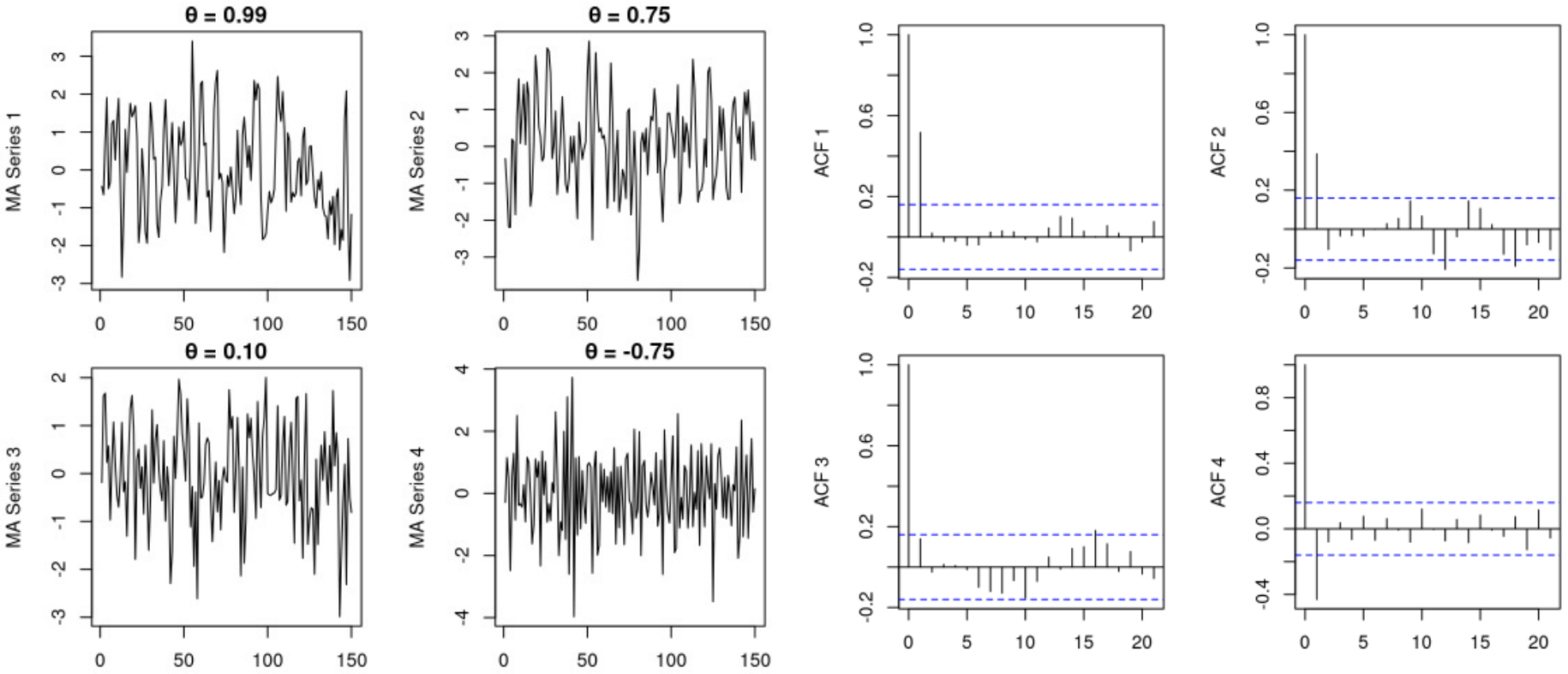
In a moving average (MA) model, Y_t is predicted by one or more previous values of the error/residuals ϵ_t.

q = order of the MA model

$$Y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

θ = value of the autocorrelation

Moving average component



Seasonal component

In a seasonal model (sometimes called SARIMA), Y_t is predicted by one or more previous values of Y_{t-s} at a regular interval (s, the season). Seasonality is dealt with through differencing, not inclusion of seasonal dummy variables in the model.

$$Y_t = c + \Phi Y_{t-s}$$

s = season

Φ = value of the autocorrelation

ARIMA notation

$$ARIMA(p, d, q)$$

p = the order of the AR part of the model;

d = the differencing needed to eliminate trend (almost always 0 or 1);

q = the order of the MA part of the model;

p, d, q, are all integers, and can also be zero.

ARIMA notation

A seasonal ARIMA model is indicated by:

$$ARIMA(p, d, q) \times (P, D, Q)$$

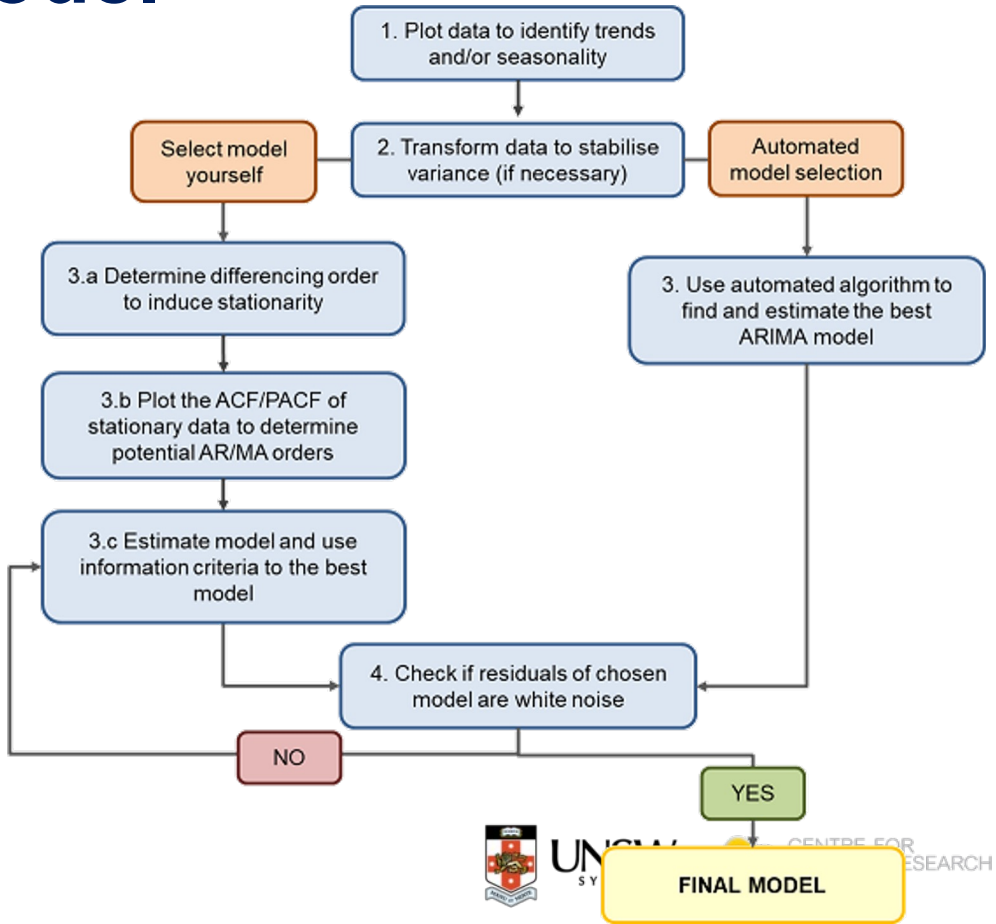
Where P and Q are the autoregressive and moving average orders for the seasonal component, and D is the degree of seasonal differencing (almost always 0 or 1).

If $d = 1$ and your data are seasonal, then D will almost always be 1 as well.

Fitting an ARIMA model

The best approach is to specify *d* and *D,* and let an automated algorithm select the best terms for *p* and *q* (and *P* and *Q* if data are seasonal), like `auto.arima` in the *forecast* package.

Always check that your final model has a good fit!



auto.arima syntax

```
auto.arima(y=, d=, D=, xreg=,
   seasonal=c(TRUE,FALSE),
   stepwise=c(TRUE,FALSE))
```

```
Series: z
ARIMA(3,0,1) with zero mean

Coefficients:
         ar1     ar2      ar3     ma1
      0.5317  0.6097  -0.5779  0.9771
s.e.  0.0645  0.0717   0.0635  0.0207

sigma^2 estimated as 0.9394:  log likelihood=-277.12
AIC=564.24   AICc=564.55   BIC=580.73
```

For a **seasonal** time series that is **non-stationary**, typically specify:

```
auto.arima(data.ts, d=1, D=1,
   seasonal=TRUE,
   stepwise=FALSE)
```

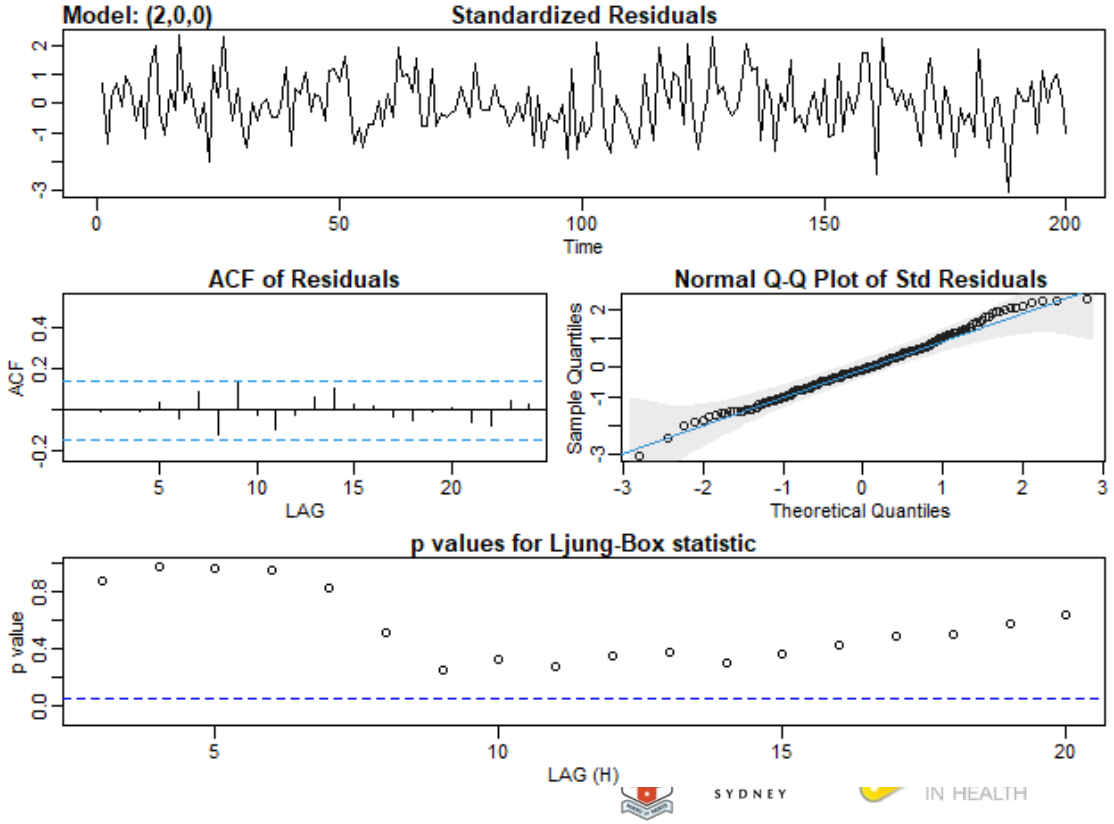
`xreg=` is for inclusion of covariates, which we will see next Chapter.

Checking model fit

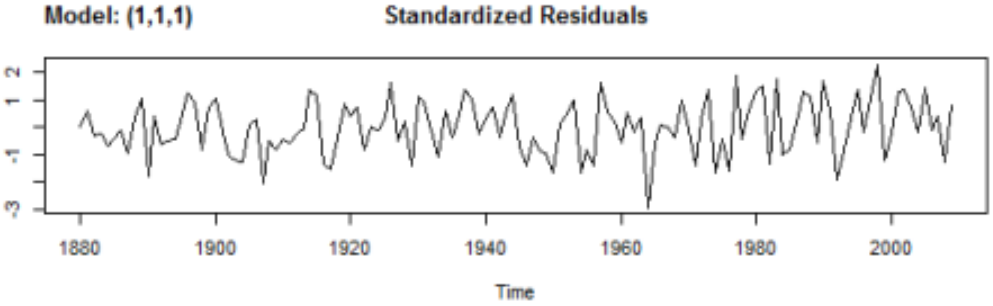
```
sarima(data=, p=, d=,
  q=, P=, D=, Q=, S=,
  xreg=)
```

You can rerun your select model in `sarima` (in the *astsa* package), which automatically generates residuals plots, and tests for autocorrelation

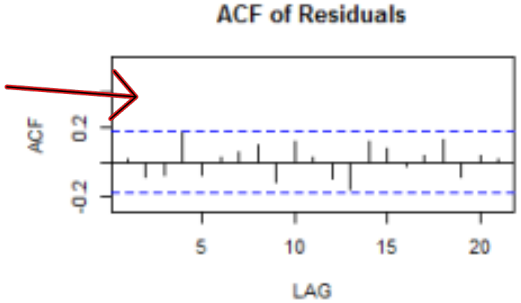
```
sarima(data.ts, p=2,
  d=0, q=0)
```

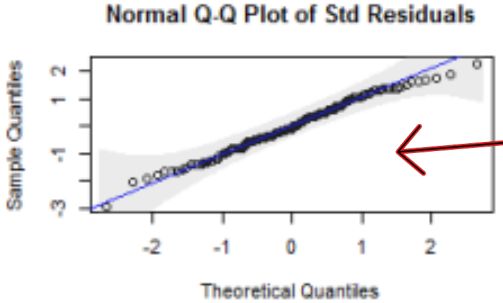


Should be stationary (constant mean and variance)



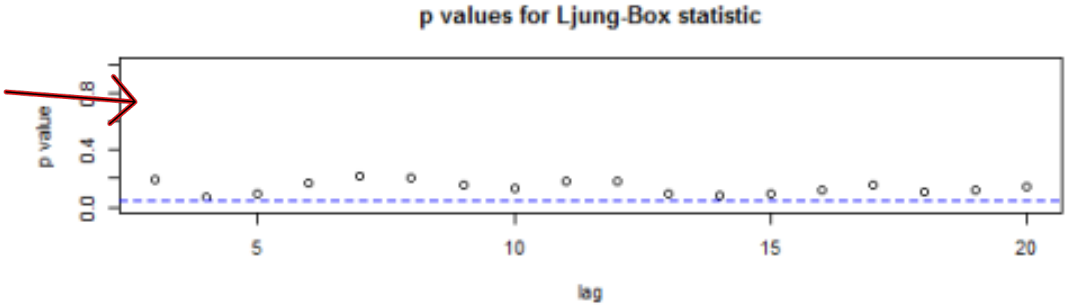
No residual autocorrelation (bars below blue line)





Normally distributed residuals (dots follow straight line)

No residual autocorrelation (dots above blue line)



What next?

Typically the model estimates for the values of the autocorrelation (AR) and/or moving average (MA) components are not of interest.

```
Series: z
ARIMA(3,0,1) with zero mean

Coefficients:
          ar1     ar2      ar3     ma1
      0.5317  0.6097  -0.5779  0.9771
s.e.  0.0645  0.0717   0.0635  0.0207

sigma^2 estimated as 0.9394:  log likelihood=-277.12
AIC=564.24   AICc=564.55   BIC=580.73
```

Once the best model is chosen, this can then be used as a base to add predictors using `xreg=` (to be covered in Chapter 5), or for forecasting.

Final words

Before embarking on any analysis of time series data, make sure you understand the patterns in your data, as well as the processes that have created it.

Aside from statistical issues, it is also important to understand how data were generated and recorded. Has this changed over time? What is captured/not captured?

While this Chapter has introduced you to several concepts related to time series data and analysis, in the next Chapter we will put this into practice.